非線形ナップサック型問題を解くための標的アプローチ (あいまいさと不確実性を含む状況の数理的意思決定)

伊佐田, 百合子; James, Ross J.W.; 仲川, 勇二

数理解析研究所講究録 1252 巻 2002 年 53-59

非線形ナップサック型問題を解くための標的アプローチ

関西大学大学院総合情報学研究科博士課程後期課程　伊佐田百合子(Isada Yuriko)
Graduate School of informatics, Kansai University
Department of Management, University of Canterbury　Ross J.W. James
関西大学総合情報学部　仲川勇二(Yuji Nakagawa)
Department of Informatics, Kansai University

1. はじめに

多目的ナップサック問題を解くために動的計画法や分枝限定法をもとにしたアルゴリズムが開発されてきた[1]. しかしこれらの多くは, 線形なナップサック問題を扱うものであった. また, 多次元非線形ナップサック問題においては, これらの手法は有効ではなく, ヒューリスティックな解法が用いられることが多かった.

本論文は, 非線形ナップサック型問題を解くために提案されているアルゴリズム（標的アプローチ）の改善を提案する. 標的アプローチは, 仲川によって開発されたモジュラ法[2]によってアルゴリズムが設計された非線形ナップサック問題の標的値より良いすべての解を制約条件のもとに列挙する手法である. 標的値より良いすべての解を列挙する問題を標的問題と呼び, そこで得られた解を標的解と呼ぶ. 次に, 単一制約単一目的の最も簡単な標的問題を示す.

$$
\begin{aligned}
&[\mathrm{P}^{\mathrm{O}}(f^{T}, b^{\mathrm{O}})]: \text{Enumerate all solutions hitting} \\
&\text{a target} \quad f^{\mathrm{O}}(\mathbf{x}) = \sum_{i=1}^{n} f_i^{\mathrm{O}}(x_i) \geq f^{T} \\
&\text{subject to} \quad g^{\mathrm{O}}(\mathbf{x}) = \sum_{i=1}^{n} g_i^{\mathrm{O}}(x_i) \leq b^{\mathrm{O}} \\
&\qquad\qquad x_i \in K_i^{\mathrm{O}} \ \text{for } i \in N,
\end{aligned}
$$

ここで, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, $K_i^{\mathrm{O}} = \{1, 2, \ldots, k_i^{\mathrm{O}}\}$, $N = \{1, 2, \ldots, n\}$, f^T は標的値である.

仲川により提案された代理制約法[3]は, 双対ギャップを閉じるために標的問題を用い, 例えば, 3 制約 1000 変数で各変数に対する項目が 20 の大規模なナップサック問題において, 厳密解を得ることに成功している. 宮地ら[4]は多目的離散最適化問題を解くために, それぞれの目的関数に標的を与え解を列挙することを提案した. 仲川ら[5]は, 多目的離散最適化問題の各目的関数に対する標的問題を代理目的乗数を用いて単一の標的問題に変換し多目的離散最適化問題を解くことを提案した. 本論文では, 多目的ナップサック問題の複数の目的関数を代理目的乗数を用いて単一の目的関数に変換した上で標的値より良い解を列挙することを検討する. さらに, 標的値の決定方法の改善を行い, 改善された手法を用いて従来の方法では困難であった大規模な問題を解くことを試み, 実際問題に適用可能であることを検証す

2. モジュラ法

モジュラ法は，単一制約で変数分離可能な問題を効率よく解くことができ，動的計画法と同様にボトムアップ的な手法である．まず，与えられた離散最適化問題に対応した最適化システムを考え，次の(1)と(2)の操作を繰り返すことにより原問題を解く．

(1) システムに対して深測操作を適用し決定空間を縮小する．

(2) システム内の複数のモジュールを単一のモジュールに統合することによってモジュール数を減らす．

(1)では，実行可能性操作，限界値操作，優越性操作の3つの深測操作を用いて，決定空間を縮小する．ただし，多目的問題を解く場合には，優越性操作は行わない．これらの深測操作は，動的計画法，及び，分枝限定法で用いられている方法と同じ手法である．モジュラ法は，Marsten－Morin[6]によって提案された動的計画法に分枝限定法の限界値操作を導入したハイブリッドDP/B&Bを更に拡張した方法である．(2)において複数のモジュールを単一のモジュールに統合するとは，2つの変数の解空間の直積を求め，その直積空間と対応する解空間をもつ新しい変数で元の2つの変数を代替することである．動的計画法において変数の統合順序が固定的であることに比較して，モジュラ法では，統合する変数を柔軟に決定できる．モジュラ法を多重選択ナップザック問題に適用した論文[7]では，変数の統合政策を適切に選択することにより実用レベルの問題に対して有効であることが示されている．非線形計画問題に適用した論文[8]では，非線形計画問題を離散最適化問題に変換した上で，モジュラ法を適用することによって，原問題を解きうることが示されている．

3. 標的アプローチの改善

標的アプローチは，問題に対して標的値を設定し，その値より良いすべての実行可能解を列挙する手法である．標的値を変化させることにより，任意の大きさの解集合を探索することができる．しかし，必要とする大きさの解集合を得るために必要とされる標的値が何かをあらかじめ知ることはできない．従来の手法では，標的値をさまざまに変化させ試行錯誤しながら，必要とする大きさの解集合が得られるまで繰り返し問題を解く必要があった．そのため，問題の規模が大きくなれば適切な標的値の特定も困難となり，実際の問題に適用するのは非常に困難であった．そこで，標的値を探索するために上限値，下限値および列挙したい解の個数を与え，2分探索を行うことにより，自動的に標的値を獲得する方法を検討する．単一制約単一目的の最も簡単な標的問題P0を用いて以下にそのアルゴリズムを示す．

（ステップ1）問題P^0をモジュラ法を用いて解く．

（ステップ2）ステップ1で得られた最適値を標的値の上限に，標的値の下限に任意の値を設定し，列挙したい解の個数を設定する．

（ステップ2）問題P^0をモジュラ法を用いて解き，得られた解の個数が列挙したい解の個数に一致するまで標的値の下限値を更新しながら，標的値の上限値と下限値の

間を２分探索する．

（ステップ３）ステップ２で得られた解の個数が列挙したい解の個数に一致していれば，ステップ２で設定した標的値の下限値を標的値に決定し探索を終了する．得られた解の個数と列挙したい解の個数を比較して得られた解の個数が少なければ標的値の下限値を緩め，多ければ下限値を厳しく再設定し，ステップ２に戻る．

4. 一次元多目的非線形ナップサック問題への適用

4.1. 代理目的問題

一次元多目的ナップサック問題は次のように定式化される．

$$
(\mathrm{P}^1):\quad \text{Maximize}\quad \mathbf{f}(\mathbf{x})=\left(f_1(\mathbf{x}),\ f_2(\mathbf{x}),\ldots,\ f_m(\mathbf{x})\right)
$$
$$
\text{subject to}\quad g(\mathbf{x})=\sum_{i=1}^{n} g_i(x_i)\le b,
$$
$$
\mathbf{x}\in\mathbf{K}
$$

ここで，$\mathbf{f}(\mathbf{x})=\left(f_1(\mathbf{x}),\ f_2(\mathbf{x}),\ldots,\ f_m(\mathbf{x})\right)$ は目的関数，$\mathbf{g}(\mathbf{x})$ は制約関数，$\mathbf{x}=(x_1,\cdots,x_n)$ は変数，$\mathbf{K}=\{\mathbf{x}: x_i\in k_i,\ for\ i=1,2,\cdots,n\}$ は，項目案集合である．多目的ナップサック問題のすべての目的関数を同時に最大化（または，最小化）するような解は存在しない．目的関数間に競合関係が存在することが一般的である．そこで，多目的ナップサック問題の解として，パレート最適解が定義されている [9]．パレート最適解は，少なくとも 1 つの目的関数の値において，他よりも優れているような解である．

問題 P^1 に代理目的乗数を導入し，複数の目的関数を単一目的関数に変換した問題を考える．この問題を代理目的問題と呼ぶ．代理目的問題は次式で表すことができる．

$$
(\mathrm{P}^2):\quad \text{Maximize}\quad \mathbf{wf}(\mathbf{x})
$$
$$
\text{subject to}\quad g(\mathbf{x})\le b,
$$
$$
\mathbf{x}\in\mathbf{K},
$$
$$
\text{但し}\quad \sum_{j=1}^{m} w_j=1,\ w_j\ge 0.
$$

ここで，$\mathbf{w}=(w_1, w_2,\ldots, w_m)$は代理目的乗数である．代理目的問題の最適解は，問題 P^1 のパレート最適解であることが知られている．

4.2. 代理目的問題への標的アプローチの適用

代理目的問題に標的アプローチを適用した問題は，次式で表すことができる．

$$
(\mathrm{P}^3):\quad \text{Target}\quad \mathbf{wf}(\mathbf{x})\ge f^{ST}
$$
$$
\text{subject to}\quad g(\mathbf{x})\le b,
$$
$$
\mathbf{x}\in\mathbf{K}.
$$

ここで，f^{ST} は代理目的問題の標的値である．この問題は，単一制約単一目的の標的問題になっている．この標的問題を解いて得られる解集合は，次の定理により，問題 $\mathbf{P}^1$ のパレート最適解である[10].

［定理］

$\mathbf{X}^T$ が標的問題 $\mathbf{P}^3$ の実行可能解集合，$\mathbf{X}^C$ は $\mathbf{X}^T$ 内で優越操作をして得られた解集合とする．

$\mathbf{X}^C = \{\mathbf{x} \in \mathbf{X}^T \mid f(\mathbf{x}^1) \geq f(\mathbf{x})$ となる $\mathbf{x}^1 \in \mathbf{X}^T$ が存在しない$\}$

とすれば，$\mathbf{x}^C \in \mathbf{X}^C$ は，原問題のパレート最適解である．

［証明］

$\mathbf{x}^1$ が問題 $\mathbf{P}^1$ の実行可能解であるとすると，以下の式が成り立つ．

$$\mathbf{x}^1 \in \mathbf{X} = \{\mathbf{x} | g(\mathbf{x}) \geq b\}$$

$\mathbf{x}^1 \in \mathbf{X}^T$ ならば，$\mathbf{X}^C$ の定義により，$f(\mathbf{x}^C) \leq f(\mathbf{x}^1)$ となるような $\mathbf{x}^1$ は存在しない．もし，$\mathbf{x}^1 \notin \mathbf{X}^T$ ならば，問題 $\mathbf{P}^3$ のパレート最適解集合は以下のように表される．

$$\mathbf{wf}(\mathbf{x}^C) \leq f^{ST} \leq \mathbf{wf}(\mathbf{x}^1)$$

しかし，$f(\mathbf{x}^C) \leq f(\mathbf{x}^1)$ となるような $\mathbf{x}^1$ は存在しない．したがって，$\mathbf{x}^C$ は $\mathbf{P}^1$ のパレート最適化解集合である．

（証明終）

多目的ナップサック問題では，得られたパレート最適解の中から意思決定者が自らの価値観にあった解を選択する．こうして得られた解は，選考最適解と呼ばれる．意思決定者が解を選択する際に，すべてのパレート最適解が列挙されている必要はない．適切な個数の意思決定者の価値観を反映した解が，解の候補として列挙されていればよい．多目的ナップサック問題に標的アプローチを適用する場合は，多目的ナップサック問題に代理目的乗数を導入して単一目的の代理目的問題に変換した上で，代理目的問題に対して標的値より良い実行可能解を列挙する．意思決定者の価値観は代理目的乗数によって与えられ，解候補の数は，標的値で与えられる．意思決定者は，改良された標的アプローチを用いて多目的ナップサック問題を解き，得られた実行可能解に対し原問題の優越性操作を行って得られるパレート最適解の中から，最も自らの価値観に合った解を選択すればよい．以下に標的問題 $\mathbf{P}^3$ を用いて，標的アプローチを多目的ナップサック問題に適用した場合のアルゴリズムを示す．

（ステップ１）問題 $\mathbf{P}^2$ をモジュラ法を用いて解く．

（ステップ２）ステップ１で得られた最適値を標的値の上限に，標的値の下限に任意の値を設定し，列挙したい解の個数を設定する．

（ステップ２）問題 $\mathbf{P}^3$ をモジュラ法を用いて解き，得られた解の個数が列挙したい解の個数に一致するまで標的値の下限値を更新しながら，標的値の上限値と下限値の間を２分探索する．ここで，実行可能解が得られなければ，更新された標的値の下限値を上限値に設定して探索を繰り返す．

（ステップ３）ステップ２で得られた解の個数が列挙したい解の個数に一致していれば，ステップ２で設定した標的値の下限値を標的値に決定し，ステップ４に進む．

列挙したい解の個数と比較して得られた解の個数が少なければ標的値の下限値を緩め，多ければ下限値を厳しく再設定して，ステップ２に戻る．

（ステップ４）得られた実行可能解に対して原問題の目的関数の優越性操作を実施し，パレート最適解を得る．

5. 計算実験

１次元１目的非線形ナップサック問題に標的アプローチを適用した場合と１次元多目的非線形ナップサック問題に適用した場合について計算実験を行った．１次元１目的非線形ナップサック問題の計算結果を表 1 に示す．１次元多目的非線形ナップサック問題に適用した計算例として，２目的の場合の計算結果を表 2に，目的の場合の計算結果を表 3に示す．それぞれ変数と各変数に対する項目案数をそれぞれ 100，200，500，1000 と変化させた問題について，列挙する解の個数を 10 個以内に設定して計算実験を行った．テスト問題は，擬似乱数 $1 \leq f_{ij}(k+1) - f_{ij}(k) \leq 2^8$ を使用して生成した．測定した項目は，アルゴリズムが停止した際の解の最大数と処理時間及び 10 個以内の解を列挙するまでの繰り返し回数である．計算実験には，ＣＰＵ500MHz，メモリー192MB の DOS/Ｖコンピュータを使用した．計算実験の結果から，1000 変数，1000 項目案の大規模な問題であっても，一回の解の探索は 10 分～20 分で完了しており，希望する解のサイズになるまで繰り返し探索した場合でも 1 時間から 1.5 時間程度で解を求め得る．このことから，現在まで解くことが困難であった大規模な問題をパーソナルコンピュータにおいて実用的な時間内で解きうることを示している．

表 1　1 目的，100，200，500，1000 変数で 100，200，500，1000 項目案の場合の計算

$n \times k^0_i$	10 個以内の解列挙時		総処理時間(sec)	繰り返し回数
	最大項目数	処理時間(sec)		
100×100	12266	0	4	8
100×200	55356	2	8	5
100×500	53656	4	41	10
100×1000	13024	6	59	10
200×100	15120	2	14	6
200×200	1712	4	34	8
200×500	128432	20	131	8
200×1000	n/k	n/k	n/k	n/k
500×100	5374	15	119	8
500×200	35602	31	236	8
500×500	12803	67	134	8
500×1000	21027	126	1009	8
1000×100	16925	64	125	2
1000×200	88323	114	677	6
1000×500	64449	240	972	4
1000×1000	10983	505	3529	7

n/k : not known.

表 2　2 目的，100，200，500，1000 変数で 100，200，500，1000 項目案の場合の計算結果

$n \times k^0_i$	10 個以内の解列挙時		総処理時間(sec)	繰り返し回数
	最大項目数	処理時間(sec)		
100×100	54411	3	9	5
100×200	6193	2	9	5
100×500	1562	4	38	9
100×1000	9728	9	56	6
200×100	15180	4	32	8
200×200	1588	6	52	8
200×500	263336	26	125	5
200×1000	934416	59	435	8
500×100	90226	29	190	7
500×200	10111	42	288	7
500×500	43771	101	202	2
500×1000	344058	241	957	4
1000×100	28327	99	594	6
1000×200	9175	171	1187	7
1000×500	328994	452	2625	6
1000×1000	35686	777	5431	7

表 3　3 目的，100，200，500，1000 変数で 100，200，500，1000 項目案の場合の計算結果

$n \times k^0_i$	10 個以内の解列挙時		総処理時間(sec)	繰り返し回数
	最大項目数	処理時間(sec)		
100×100	2925	1	5	4
100×200	24660	4	24	8
100×500	8708	6	37	6
100×1000	97022	16	45	3
200×100	6758	5	30	6
200×200	1920	9	44	5
200×500	964641	76	400	7
200×1000	189890	57	302	5
500×100	28728	34	130	4
500×200	10679	56	335	6
500×500	235494	152	1201	8
500×1000	55563	291	1745	6
1000×100	13481	133	268	2
1000×200	47301	235	1161	5
1000×500	71273	552	3907	7
1000×1000	22323	1015	4173	4

6. おわりに

単一制約の多目的非線形ナップサック問題において，標的値の最適化を自動的に行うことにより，従来方法と比較して効率的にパレート最適解を求める方法を示した．計算機実験に

より，この手法を用いて1000変数で1000項目案数の規模の大規模な非線形ナップサック型の問題が解き得ることを示した．この手法を用いて得られた解は，意思決定者の価値観を反映した意思決定者が必要とするサイズの解集合になっている．意思決定の現場では，多目的非線形ナップサック問題を解いて得られたパレート最適解の中から意思決定者が解を選択する必要があるので，意思決定者の価値観にあった適切なサイズのパレート解集合が効率的に提示できることは非常に重要である．本論文では，標的値を操作することで得られる解のサイズを最適化し効率的に解を得る方法を提案した．今後の課題として，意思決定者の価値観を反映するために目的乗数を最適化する手法の検討が必要である．

参考文献

[1] M. Ehrgott, X.Gandibleux:"A survey and annotated bibliography of multiobjective combinatorial optimization",OR Spektrum 22, pp425-460(2000).
[2] 仲川勇二:「離散最適化問題のための新解法」,電子情報通信学会論文誌,Vol. J73-A, No. 3, pp. 550-556(1990).
[3] Y. Nakagawa:"A reinforced surrogate constraints method for separable nonlinear integer programming", RIMS Kokyuroku 1068 of Kyoto University, Nov, pp.194-202. (1998).
[4] I. Miyaji, Y. Nakagawa: "Decision support system for the composition of the examination problem", European Journal Of Operational Research80 pp.130-138(1995).
[5] 仲川，疋田：「多目的離散最適化問題のための対話型意思決定アルゴリズム」，経営工学会論文誌，Vol. 51，pp. 197-202（2000).
[6] R.E.Marsten, T.L.Morine:" A hybrid approach to discrete mathematical programming", Math. program, Vol14, pp.21-40(1974).
[7] 仲川勇二，疋田光伯，岩崎彰典，「多重選択ナップザック問題の高速厳密解法」，電子情報通信学会論文誌，Vol. J75-A，No. 11，pp. 1752-1754（1992).
[8] 疋田光伯，岩崎彰典，仲川勇二，「モジュラ法の非線形計画問題への適用」，電子情報通信学会論文誌，Vol. J76-A，No. 1，pp. 64-67，Jan. 1993.
[9] 西川，三宮，茨木：「岩波講座情報科学19最適化」，岩波書店，pp. 162-166（1982).
[10] Y.Isada, M.Hikita, Y.Nakagawa: "A Method for Solving Multi-objective Discrete Optimization Problem", Proceedings of the first western pacific and third australia-japan workshop on stochastic models in engineering, technology and management, pp.192-201 (1999)